

ПРИМЕНЕНИЕ МИКРОКОНТРОЛЛЕРА ОБЩЕГО НАЗНАЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРОДВИГАТЕЛЯМИ ПОСТОЯННОГО ТОКА

Филиал «Протвино» университета «Дубна»

Кафедра автоматизации технологических процессов и производств

Целью данной работы является ознакомление, разработка, изготовление, наладка и демонстрация макета опытной установки, состоящей из электродвигателя и его микроконтроллерной системы управления и их прикладное применение в различных сферах промышленности и робототехники.

Электродвигатели постоянного тока широко используются в разного рода технических устройствах. Например: следящие приводы, сложные металлообрабатывающие станки, сервоприводы и т.д. Рассмотрим схему устройства управления двигателем постоянного тока (ДПТ) мощностью до 100 ватт (рис. 1).

Она состоит из 8-битного микроконтроллера производства *Microchip* семейства *PIC18F* модель 452, интегрального драйвера *ST L298* мощностью до 150 ватт. Рассмотрим подробно элементы данного изделия и алгоритм работы данного устройства. Микроконтроллер подает два ШИМ (широтно-импульсная модуляция) сигнала на входы драйвера двигателя, изменяя ширину импульсов заданной частоты. При этом изменяется среднее значение питающего ДПТ напряжения и, как следствие, его скорость.

Для реализации такого алгоритма работы нам необходимо создать программное обеспечение.

Подавая на входы напряжение высокого или низкого уровня, микроконтроллер включает или выключает схему моста драйвера. Пользователь может задавать длину импульса, используя соответствующие элементы управления, расположенные непосредственно на плате устройства.

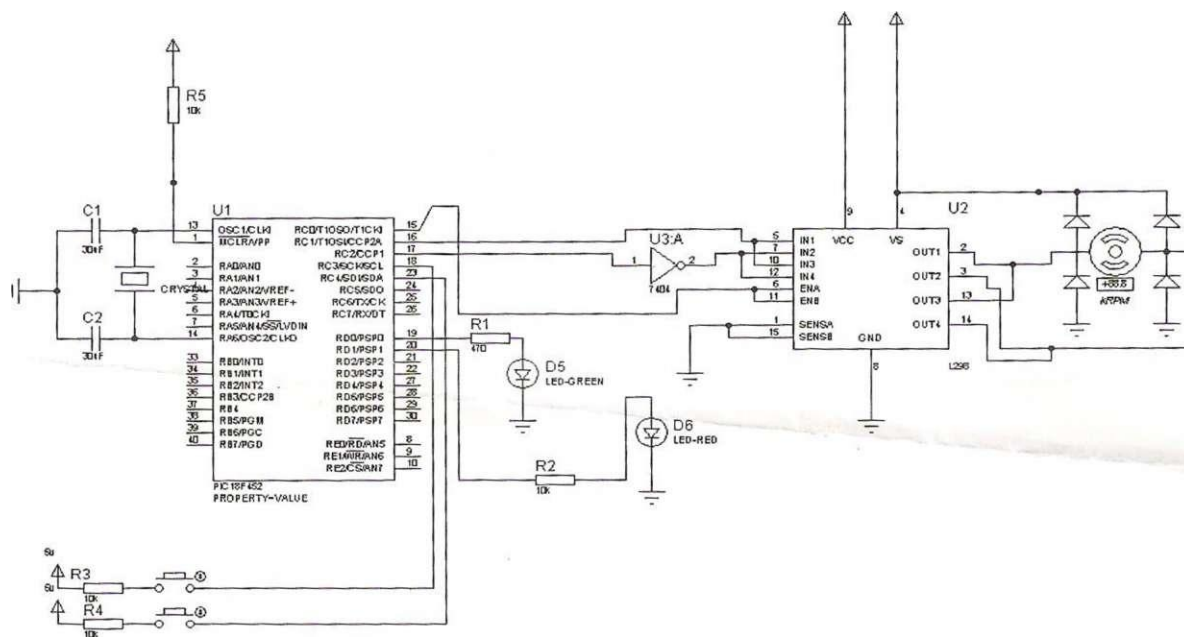


Рис. 1. Принципиальная схема устройства для управления ДПТ

А сейчас рассмотрим процессы, происходящие в данном устройстве. Драйвер двигателя представляет собой два Н-моста на транзисторах, управляемых ТТЦТ (транзисторно-транзисторной логикой). Функциональная схема его приведена на рис. 2.

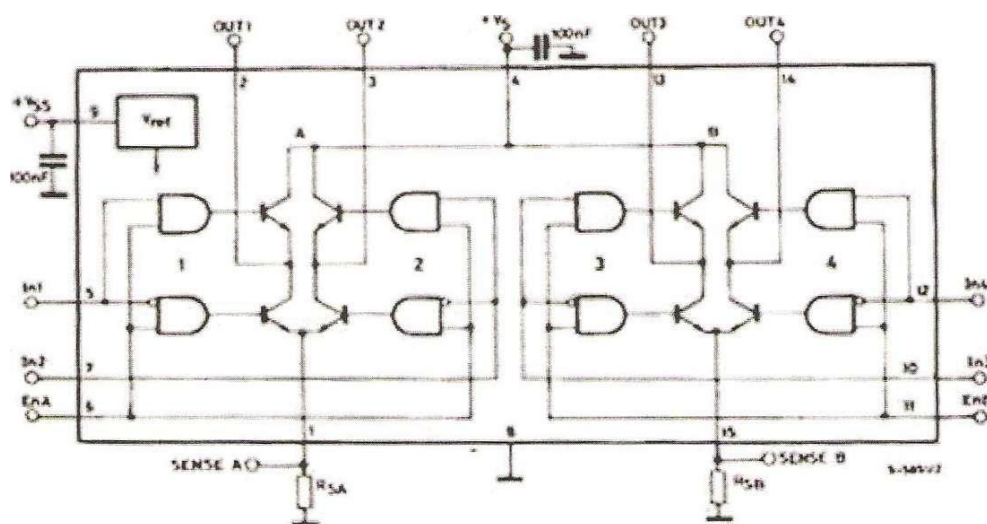


Рис. 2. Функциональная схема драйвера ДПТ

Перед каждым транзистором стоит логический элемент И. При подаче логической единицы, входы разрешения активируют соответствующий ему мост. Если подать высокий уровень на входы IN1 или IN2, то двигатель будет вращаться, соответственно, по или против часовой стрелки.

Рассмотрим таблицу 1 истинности для одного моста и состояние двигателя при разных логических уровнях на входах драйвера.

Таблица 1

Таблица состояний драйвера

Еп (Вход разрешения)	In1	In2	Функция
Высокий уровень	Высокий уровень	Низкий уровень	Вращение по часовой стрелке
Высокий уровень	Низкий уровень	Высокий уровень	Вращение против часовой стрелки
Высокий уровень	Низкий уровень	Низкий уровень	Остановка двигателя
Низкий уровень	Низкий/Высокий уровень	Низкий/Высокий уровень	Свободное вращение двигателя

Скорость вращения двигателя меняется изменением ширины импульса на входах *In1* и *In2*. Подача ШИМ на входы *In1* и *In2* дает более стабильную характеристику на выходе интегральной схемы,

Один мост рассчитан на ток в 2 ампера, а если они запараллелены, то выдерживают ток силой до 4 ампер. Так как, в данной интегральной схеме отсутствуют внутренние защитные диоды, перед выходами мостов стоят 4 высокочастотных диода *HER 307* с максимальным током в 3 ампера для его отвода в случае возникновения противо-эдс при остановке двигателя.

Выходы драйвера *Sense 1* и *Sense 2* служат для контроля тока. В случае превышения программно заданного лимита, происходит отключение питания электродвигателя.

В нашем устройстве они не задействованы, а просто заземлены.

Рассмотрим исходный код и алгоритм программы управления применительно к нашему устройству.

Программное обеспечение для микроконтроллера написано на языке *C* в фирменной среде *MPLAB* и компиляторов *MicroC* и *Microchip C18*.

Для ознакомления рассмотрим простую реализацию ПО устройства и в дальнейшем подробно на примере кода, написанного на компиляторе *Microchip*.

Исходный код:

```
unsigned char duty=127; //Сквозность 50%
```

```
void main() {
```

```
    TRISC.F0=0; //Режим пина 0 порта C выход TRISC.F1=0; //Режим пина 1 порта C выход
```

```
    TRISC.F3=1; //Режим пина 3 порта вход TRISC.F4 = 1; //Режим пина 4 порта C вход TRISC.F0 = 0;
```

```

PORTC.F0=1;//На пине 0 порта C высокий уровень PORTD.F0=1; PWM1_Init(10000);
PWM2_Init. (10000) ;
//Настройка модуля CCP на режим ШИМ с частотой 10 кГц PWM1_Start(); PWM2_Start();
do //Зацикливание для опроса кнопок
{if(Button(PORTC,3,1,1) & (duty<255))
dty=dty+16;
if(Button(PORTC,4,1,1) & (duty>0))
dty=dty-16;
PWM1_Change_Duty(dty); PWM2_Change_Duty(dty);
Delay_1sec(); }while(1);
}

```

Как видно из кода, устройство функционирует довольно просто и понятно любому пользователю.

Выводы:

1. Нашей основной задачей было научиться программировать микроконтроллер и управлять с его помощью скоростью двигателя.
2. Создан макет этого устройства, проверена его работа.

Библиографический список

1. Шпак Ю. А. Программирование на языке C для AVR и PIC микроконтроллеров / Ю. А Шпак — Изд. : Корона-Век, МК-Пресс 2011 г. — 544 с.
2. Документация к драйверу STL 258 Datasheet [www. microe.com](http://www.microe.com).
3. www.microe.com